

## Amendments to the Claims

1. (currently amended) A computer-implemented method for reverting a process in an in-line instrumented state to an uninstrumented state, said method comprising:

modifying selected text segment portions from said process which has been instrumented;

unmapping instrumented code space such that said instrumented code space is inaccessible to said process; and

cleaning a call stack of said process by unwinding said call stack and resetting a storage location for a return pointer from said instrumented code space to uninstrumented code space wherein said cleaning is performed on said process which has been instrumented, and

comparing the return addresses of the process to the address in the instrumented code space which generated a fault upon execution of said process when the process generates the fault by seeking to access an address in instrumented code space.

2. (original) The computer-implemented method for reverting a process in an in-line instrumented state to an uninstrumented state as recited in claim 1 wherein said instrumented code space is comprised of shared memory or memory in an address space of said process.

3. (original) The computer-implemented method for reverting a process in an in-line instrumented state to an uninstrumented state as recited in claim 1 wherein the step of resetting said storage location for said return pointer comprises:

finding in said call stack said storage location of said return pointer; and

changing the value of said storage location of said return pointer from said instrumented code space to uninstrumented code space.

4. (original) The computer-implemented method for reverting a process in an in-

line instrumented state to an uninstrumented state as recited in claim 1 wherein said return pointer is disposed in a renamed register.

5. (original) The computer-implemented method for reverting a process in an in-line instrumented state to an uninstrumented state as recited in claim 1 wherein said return pointer is disposed in a memory location on said call stack.

6. (original) The computer-implemented method for reverting a process in an in-line instrumented state to an uninstrumented state as recited in claim 1 wherein said return pointer is disposed in a preserved register.

7. (original) The computer-implemented method for reverting a process in an in-line instrumented state to an uninstrumented state as recited in claim 1 wherein said return pointer is disposed in an alternate branch register.

8. (original) The computer-implemented method for reverting a process in an in-line instrumented state to an uninstrumented state as recited in claim 1 wherein said return pointer is not yet saved.

9. (original) The computer-implemented method for reverting a process in an in-line instrumented state to an uninstrumented state as recited in claim 1 wherein said return pointer is disposed in a memory location of an operating system of said process.

10. (currently amended) A computer-readable medium embodying instructions that cause a computer to perform a method for reverting a process in an in-line instrumented state to an uninstrumented state, the method comprising:

modifying selected text segment portions from said process which has been instrumented;

unmapping instrumented code space such that said instrumented code space is inaccessible to said process; and

cleaning a call stack of said process by unwinding said call stack and resetting a storage location for a return pointer from said instrumented code space to uninstrumented code space wherein said cleaning is performed on said process which has been instrumented, and

comparing the return addresses of the process to the address in the instrumented code space which generated a fault upon execution of said process when the process generates the fault by seeking to access an address in instrumented code space.

11. (original) The computer-readable medium of claim 10 wherein said instrumented code space is comprised of shared memory or memory in an address space of said process.

12. (original) The computer-readable medium of claim 10 wherein said instructions further cause said computer to: find in said call stack said storage location of said return pointer; and change the value of said storage location of said return pointer from said instrumented code space to uninstrumented code space.

13. (original) The computer-readable medium of claim 10 wherein said return pointer is disposed in a renamed register.

14. (original) The computer-readable medium of claim 10 wherein said return pointer is disposed in a memory location on said call stack.

15. (original) The computer-readable medium of claim 10 wherein said return pointer is disposed in a preserved register.

16. (original) The computer-readable medium of claim 10 wherein said return pointer is disposed in an alternate branch register.

17. (original) The computer-readable medium of claim 10 wherein said return pointer is not yet saved.

18. (original) The computer-readable medium of claim 10 wherein said return pointer is disposed in a memory location of an operating system of said process.

19. (currently amended) An apparatus for reverting a process in an in-line instrumented state to an uninstrumented state, the apparatus comprising:

means for modifying selected text segment portions from said process which has been instrumented;

means for unmapping instrumented code space such that said instrumented code space is inaccessible to said process; and

means for cleaning a call stack of said process by unwinding said call stack and resetting a storage location of a return pointer from said instrumented code space to uninstrumented code space wherein said cleaning is performed on said process which has been instrumented, and

comparing the return addresses of the process to the address in the instrumented code space which generated a fault upon execution of said process when the process generates the fault by seeking to access an address in instrumented code space.

20. (original) The apparatus of claim 19 for reverting a process in an in-line instrumented state to an uninstrumented state wherein said instrumented code space is comprised of shared memory or memory in an address space of said process.

21. (previously presented) The apparatus of claim 19 for reverting a process in an in-line instrumented state to an uninstrumented state wherein said means for cleaning up said call stack further comprises:

means for finding in said call stack said storage location of said return pointer; and

means for changing the value of said storage location for said return pointer from said instrumented code space to uninstrumented code space.

22. (original) The apparatus of claim 19 for reverting a process in an in-line instrumented state to an uninstrumented state wherein said return pointer is disposed in a renamed register.

23. (original) The apparatus of claim 19 for reverting a process in an in-line instrumented state to an uninstrumented state wherein said return pointer is disposed in a memory location on said call stack.

24. (original) The apparatus of claim 19 for reverting a process in an in-line instrumented state to an uninstrumented state wherein said return pointer is disposed in a preserved register.

25. (original) The apparatus of claim 19 for reverting a process in an in-line instrumented state to an uninstrumented state wherein said return pointer is disposed in an alternate branch register.

26. (original) The apparatus of claim 19 for reverting a process in an in-line instrumented state to an uninstrumented state wherein said return pointer is not yet saved.

27. (original) The apparatus of claim 19 for reverting a process in an in-line instrumented state to an uninstrumented state wherein said return pointer is disposed in a memory location of an operating system of said process.